I. General Remarks Concerning This Response

Claims 1-89 are currently pending in the present application. The Office action has objected to claims 14, 25, 36, 47, 63, and 79 as being dependent upon a rejected based claim but otherwise allowable if rewritten in independent form. No claims have been amended, added, or canceled in this response. Reconsideration of the claims is respectfully requested.

The Office action objected to the drawings; a set of formal drawings is being mailed separately.

II. Summary of Present Invention

The present invention is a method, system, apparatus, or computer program product for mapping a source identifier in a source identifier space to a target identifier in a target identifier space using a stable hash-based computation. An information item identifiable by a source identifier is to be associated with some type of computational resource, and the computational resource is represented by a target object identifiable by one or more target identifiers. The set of target objects is dynamically variable, yet the mapping is stable over time with respect to the amount of remapping caused by a change to the set of target identifiers. After hashing the source identifier to produce an index position of an entry in a table, and a target identifier is retrieved from the table entry, thereby mapping the source identifier to the target identifier in a mapping operation whose speed is independent of the number of target identifiers.

The mapping is performed via an intermediate table, herein termed a "targetMap" table, in which each entry contains a target identifier. Other data structures could be substituted for the intermediate table, and other information associated with a target identifier may be stored in the intermediate data structure.

The targetMap table is managed as follows. Each entry in the table is related to a single target identifier, yet each target identifier may be related to more than one table entry, thereby producing a relation between a table entry and a target object. The target that is associated with a particular table entry is based on a "nearness" computation that depends upon the table index position of the particular table entry and a target identifier for the associated target. The nearness computation is performed between each table entry and each target identifier to obtain a fair distribution of relationships between table entries and targets. Targets can be added or removed with minimal impact on the table.

Target objects may have one or more associated target identifiers, the number of which is proportional to a measure of computational capacity of the target resource. Hence, the present invention also incorporates a weighting mechanism into the mapping operation such that source identifiers are mapped to target objects in proportion to the predetermined weight or capacity of the target object.

III. <u>35 U.S.C. § 103—Obviousness over O'Connell</u>

Claims 9-13, 15-24, 26-35, 37-46, 48-57, 59-62, 64-78, and 81-89 are rejected under 35 U.S.C. § 103(a) as unpatentable over Rostoker et al., "Method for hashing in a packet network switching system", U.S. Patent Number 5,708,659, filed 02/16/1995, issued 01/13/1998. This rejection is traversed.

The rejection of the claims applies various figures against the claims. In fact, the rejection of all claims uses only one long portion of <u>Rostoker et al.</u> against of the claims. Given the importance of this section, this portion of <u>Rostoker et al.</u> from column 19, line 32, to column 21, line 4, is copied hereinbelow to allow easy reference of the subject matter:

When packets of data are typically received by a hub 311 or a router 312, address information contained in the packet header must be used to determine where the packet is to be switched. Packet address information may typically comprise 48 bits, and simple algorithms for comparing such information with all known destination addresses would be impractical, because speed is critical in such applications. Hashing logic 546 may implement suitable hashing algorithms, which are used in network switching systems such as routers 312 and hubs 311 to quickly search a table of address/port combinations to determine the correct port for a specific destination address. This enables the packet to be switched to the port containing the end user to whom the packet is addressed, eliminating the need to broadcast the message across all ports in a shotgun approach. Flooding the network with broadcast packets increases traffic on the network, and slows throughput. Hashing logic 546 is used to implement fast methods for determining the correct port for a given destination address, and by avoiding the need to broadcast packets, such arrangements effectively increase the bandwidth of the network 301.

In token ring, Ethernet, FDDI and CDDI network protocols, the address of the end user is contained in 48 bits in the header of the packet. The hashing logic 546 implements a suitable hashing algorithm which in a preferred embodiment can be performed on a variable number of bits from one to all 48. A variable window can be applied to the address so that, for instance, bits three through seven may be chosen as the bits to hash. The hash does not always have to start with bit zero. Utilizing a microprocessor, a hash key size can be selected by software programming. Similarly, a "window" location, i.e., the bits selected from the 48 bits of address information which will be hashed, can be selected under software control using a microprocessor implementation.

FIG. 46 depicts a flow chart of certain steps performed during hashing. In a hash method, a key is selected from a subset of the total address bits contained in the packet header in step 800. Hash logic 546 computes an address in a table in step 801. In step 803, the computed address contents are compared to the entire destination address to identify whether there is a match. As shown in step 804, if there is a match, then the corresponding port address stored in the table has been located and the search is successful. If the contents do not match, a collision has occurred, and the hash address is incremented by one in step 805, and the comparison is performed again as described above in step 803. In this fashion collisions are allowed by having a linked list approach for overflow which allocates extra memory slots to each possible hash. If the table location

corresponding to the hash is empty, then the system will have to either (1) broadcast an error and not forward the packet since it cannot identify the port to route the packet, or (2) broadcast the packet to all of the ports.

Referring now to the flow chart shown in FIG. 47, the system learns the end user address and corresponding port from the originator of a packet. The system records the port address from the incoming message, hashes the address within the packet corresponding to the originator in step 807, looks up in the table for that particular hash address value in step 808, and compares the contents in step 811. If there is a match, the system does nothing since the end user address and port location are already stored in the table and therefore known by the system. This is shown by the step "end" in FIG. 47 associated with step 811. If the hash address is empty the system writes to that location the end user address and the port number in step 809. If the address is full but the contents are different, the system increments the hash address by one value in step 812, and proceeds to step 811 to compare the contents of that location as described above. Collisions are thus handled by allowing for overflow in the memory.

If the overflow is full and the address has not been found, the system can be programmed to write over the contents of one of the memory locations for that hash key. Extra table memory can be used to record the number of messages sent to a particular address from the hashing exercise performed on every packet. In that way, the most popular address locations could be saved in the table and the less popular addresses dropped so that the overflow memory locations for each hash address can be minimized.

In the above example, the memory size, the number of ports, the hashing algorithm, the size of the overflow list for collisions, the key size and the key window on the address are all programmable.

An example of a suitable hashing algorithm is $h(K)=K$ modulo M where h is the hash function address result, K is the key and M is a value such that $0<h(K)<M$. Since the scheme described above enables variable key sizes and key windows, the user may select the appropriate values for the key "K" and the value "M". The user may therefore wish to hash on three bits of the end user address corresponding to bits five, six and seven (note variable window). Since three bits in a digital system correspond to eight specific values, and since $h(K)<M$, M must be greater than eight. Since the best values for M to minimize collisions are odd, not multiples of three, and are a prime number, the number 11 works well. The corresponding h(K) can then be multiplied by the size of the memory table to compute the address.

> The ports 542, 543, and 544 may employ circuits disclosed in "L64380 Quad CASCADE Technical Manual" (Dec. 2, 1994) available from LSI Logic. Considerable circuits are shown in more detail in FIG. 34 and FIG. 35. FIG. 35 shows an external MAC interface 557. FIG. 33 shows an alternative embodiment of the multiport switch shown in FIG. 29, including a DMA controller 558.

The rejection analyzes the terms "source identifier", "target identifier", and "table index" within claim 9 as analogous to "packet address", "address information", and "key" as disclosed in <u>Rostoker et al.</u> by stating: "Referring to Claim 9, Rostoker teaches: method (...); source identifier (packet address ...); target identifier (address information ...); table index (key ...); hashing (hash ...)." The rejection then states the following on page 4 of the Office action:

> Rostoker et al. does not expressly call for: target identifier has been related to the stored in the table entry [sic] based on a computed value from a relation computation using the table index and the target identifier as operands in the relation computation but teaches address info hashed to determine key index that is utilized to determine address info re-computing the address info based upon the index and address info in the event that the contents do not match per col. 19 line 32-col. 21 line 4.
>
> It would have been obvious to one of ordinary skill in the art at the time of the invention that having the address info hashed to determine key index that is utilized to determine address info re-computing the address info based upon the index and address info in the event that the contents do not match per col. 19 line 32-col. 21 line 4 performs the same function as the target identifier has been related to the stored in the table entry [sic] based on a computed value from a relation computation using the table index and the target identifier as operands in the relation computation.

Hence, the rejection admits that <u>Rostoker et al.</u> does not disclose a portion of the second element of independent claim 9, which reads in its entirety:

9. A method in a data processing system for mapping a source identifier to a target identifier, the method comprising the steps of:

hashing the source identifier to determine a table index into a table in a computer readable medium; and

reading the target identifier from a table entry using the table index, wherein the target identifier has been related to and stored in the table entry based on a computed value from a relation computation using the table index and the target identifier as operands in the relation computation.

Applicant strongly disagrees with the argument in the rejection that the steps that are disclosed or suggested in Rostoker et al. perform the same function as is claimed in the present patent application.

As an initial point, though, Applicant notes that the rejection is inconsistent in several aspects. Applicant notes that the rejection twice states the feature within Rostoker et al. that is being relied upon by the rejection for supporting its argument. However, Applicant notes that the clause "... teaches address info hashed to determine key index that is utilized to determine address info re-computing the address info based upon the index and address info in the event that the contents do not match" is confusing in each of the two instances in which it is used within the rejection. It is clear that Rostoker et al. discloses that the packet's destination address information is hashed; the destination address information is actually a hash key that is selected from a subset of the total address bits in the destination address. The hash key is used to compute a table address, and the table address is used to access the table to retrieve the destination address that is stored as part of the contents at the computed table address, i.e. the table entry at that table address: "Hash logic 546 computes an address in a table in step 801. In step 803, the computed address contents are compared to the entire destination address to identify whether there is a match."--(Rostoker et al., column 20, lines

4-7). These two steps, steps 801 and 803, appear to be the steps that are referred to by the rejection when it states "... teaches address info hashed to determine key index that is utilized to determine address info". After this portion, it is unclear whether the rejection intended to use a conjunction to bring together the next part of the rejection's statement because it jumps into another verb by stating "re-computing the address info based upon the index and address info in the event that the contents do not match". Applicant believes that the rejection intended to say the following (emphasis added to denote additional words that have been added by Applicant):

> ... teaches address info hashed to determine key index that is utilized to determine address info **and then** re-computing the address info based upon the index and address info in the event that the contents do not match.

In other words, the rejection is referring to the portion of Rostoker et al. that discusses the resolution of collisions, which is significant for reasons that are discussed in more detail further below.

Applicant asserts that the rejection is inconsistent for other reasons. The rejection has either misinterpreted the teachings of Rostoker et al., misinterpreted the claim language of the present patent application, or both. For example, the rejection uses the term "key index", but this term is not used within Rostoker et al. nor within the claims of the present patent application; the term "key index" appears to be a combination of the terminology from Rostoker et al. and from the present patent application. The beginning of the rejection adds to this confusion by stating that the term "key" as used in Rostoker et al. is analogous or equivalent to Applicant's use of the term "table index". If Applicant interprets the rejection's use of the term "key index" as "table index", then it is unclear what would be meant by the rejection's analogy of "key" with "table index".

In addition, the rejection uses the term "address information" with respect to Rostoker et al. multiple times without clearly distinguishing the address information to which the rejection is referring. In the first instance, the rejection states that "address info is hashed"; the address information that is hashed is the destination address (or a subset of its bits) from the packet header. Thus, at first, the term "address info" refers to the destination address information. In the second instance, the rejection then states that the "key index" is utilized "to determine address info"; the address information that is determined within the system that is disclosed in Rostoker et al. could be one of multiple different types of address information. For example, the term "address info" could refer to: (1) an address into the hash table, such as a table index, which is disclosed in Rostoker et al. at column 20, line 4, by stating "computes an address in a table" or at column 20, line 30, by stating "If the hash address is empty, ..."; (2) the copy of the destination address that is stored in the table entry; or (3) the copy of "the corresponding port address stored in the table"--(Rostoker et al., column 20, line 8), which is stored together with the copy of the destination address in the table entry. In the third instance, it is unclear what the rejection means by the phrase "re-computing the address info"; in this instance, it seems that the hash address, i.e. the table address, seems most appropriate, but if this is the case, then it is unclear what would be meant by the rejection's use of "key index". In the fourth instance, the phrase "based upon the index and address info" would seem to require the interpretation of "address info" as the destination address again. In any case, whichever meaning is employed to interpret the use of "address info" in one instance, the rejection employs a different meaning for "address info" in a different instance.

As mentioned above, the argument in the rejection appears to rely on the portion of Rostoker et al. that discusses the resolution of collisions in the hash table as support for its argument. Moving away from the inconsistent logic in the rejection in order to focus on the disclosure of Rostoker et al., Applicant notes that claim 9 should be analyzed by the rejection in view of Rostoker et al. in the following manner.

Any system that employs a hash table uses the hash table to quickly map one set of values to another set of values. Rostoker et al. discloses the use of a hash table "to quickly search a table of address/port combinations"--(Rostoker et al., column 19, line 43). In other words, a destination address from a packet header is mapped to a port; Rostoker et al. refers to a port within the disclosed system as a "port", a "port number", a "port location", or a "port address". In a similar fashion, the present invention uses a hash table "for mapping a source identifier to a target identifier", as is recited in the preamble of claim 9. Hence, the term "source identifier" of the present invention is analogous to "destination address" in Rostoker et al., and the term "target identifier" of the present invention is analogous to "port number" or "port address" in Rostoker et al.; just as the destination address (or possibly only a portion of the destination address) is used as the hash key in Rostoker et al., the source identifier is used as the hash key in a hash function in the present invention, e.g., as recited in claim 9 by "hashing the source identifier to determine a table index ...". In addition, the term "table index" of the present invention is analogous to "computed address" or "hash address" in Rostoker et al., and the term "table entry" of the present invention is analogous to each of the many terms in Rostoker et al. that are used to refer to an entry or location within its hash table, such as "corresponding address contents", "table location", "memory slot", "contents of that location", and "address locations".

Given the proper analogies between the data items in the claims of the present invention and the data items in the system that is disclosed by Rostoker et al., the assertion in the rejection that Rostoker et al. discloses or suggests "the same function" should be re-interpreted using the proper analogies between terms. If this re-interpretation is done, the following statement from the rejection changes meaning:

> ... teaches address info hashed to determine key index that is utilized to determine address info re-computing the address info based upon the index and address info in the event that the contents do not match.

Applicant argues that the statement should be re-interpreted as follows using the appropriate terms from Rostoker et al.:

> Rostoker et al. teaches a system in which destination address information is used as a key into a hash function. The destination address information is hashed to determine a hash table location (or a hash table address) that is utilized to determine the destination address that is stored in the contents at that hash table location. If the contents are different from the destination address, i.e. in the event that the contents do not match, then there is a collision, and another hash table location is recomputed based upon the previous hash table address, which itself is based on the destination address.

Alternatively, Applicant argues that the statement should be re-interpreted as follows using the appropriate terms from the present invention:

> Rostoker et al. teaches a system in which a source identifier is used as a key into a hash function. The source identifier is hashed to determine a table index into a hash table that is utilized to determine the source identifier that is stored in the contents at that hash table location. If the contents are different from the source identifier, i.e. in the event that the contents do not match, then there is a collision, and another table index is recomputed based upon the previous table index, which itself is based on the source identifier.

In either case, when the statement from the rejection is analyzed in this manner, it should be clear that the manner in which the data items in Rostoker et al. are being used is not equivalent to

the manner in which the data items in the present invention are being used. In fact, with respect to the operations on a hash table and the manner in which collisions are resolved, Applicant asserts that Rostoker et al. only discloses or suggests typical hash table processing and typical hashing collision resolution, which Applicant attempted to distinguish from the present invention in the "Background of the Invention" section of the specification of the present patent application.

Given the proper analogies between the data items in the claims of the present invention and the data items in the system that is disclosed by Rostoker et al., it should be apparent that Rostoker et al. does not disclose nor suggest the present invention. Turning to the actual claim language in claim 9, the following feature:

> ... wherein the target identifier has been related to and stored in the table entry based on a computed value from a relation computation using the table index and the target identifier as operands in the relation computation

would require the following hypothetical feature within Rostoker et al. if analogous terminology from Rostoker et al. were substituted into the claim:

> wherein the port number has been related to and stored in the hash table location based on a computed value from a relation computation using the hash table location and the port number as operands in the relation computation.

The system that is disclosed in Rostoker et al. clearly does not use a port number to decide at which hash table location to associate, i.e. relate, and store the port number, thereby preparing the hash table location for the reading operation in the second element of claim 9.

It should also be noted that the motivational statement in the rejection of claim 9 is completely generic. There is no description of a hypothetical modification to the system that is disclosed in Rostoker et al.. In addition, the actual motivation in the rejection for modifying the system that is disclosed in

Rostoker et al. is supposedly the actual claimed feature of the present invention. It should not need mentioning that the claimed feature itself should not be the motivation; one typically looks to an advantage of a modification as being the hypothetical motivating factor, not the feature itself. Moreover, it is confusing why the motivational statement in the rejection asserts that a feature in Rostoker et al. "performs the same function"; assuming *arguendo* that a feature in Rostoker et al. "performs the same function", it is unclear why the rejection did not make an anticipation argument using Rostoker et al..

With respect to the dependent claims that depend from independent claim 9, the rejection clearly disregards most of the features of these dependent claims. For example, with respect to dependent claim 10, the rejection states that Rostoker et al. discloses the feature of "using the target identifier as input". However, claim 10 recites in its entirety:

> 10. The method of claim 9 further comprising a step of relating a particular table entry to a target identifier in which:
>
> for each target identifier in the set of target identifiers, generating a computed value using the table index for the particular table entry and a target identifier as operands in the relation computation to obtain a set of computed values;
>
> choosing a computed value from the set of computed values based upon a mathematical relationship among the set of computed values; and
>
> determining a related target identifier for the particular entry based on the chosen computed value, wherein the chosen computed value was computed using the related target identifier as input.

With respect to dependent claim 11, the rejection states that Rostoker et al. discloses the feature of "storing in a table entry it related target identifier [sic]". However, claim 11 recites in its entirety:

11. The method of claim 10 further comprising, prior to the step of reading the target identifier from the table entry:

obtaining a set of target identifiers; and

for each table entry, relating a target identifier from the set of target identifiers to a table entry such that each table entry is related with only one target identifier; and

for each table entry, storing in a table entry its related target identifier.

With respect to dependent claim 16, the rejection states that Rostoker et al. discloses the feature of "target identifiers is proportional to the computational capacity of computer resources". However, claim 16 recites in its entirety:

16. The method of claim 9 further comprising:

associating a computational resource with a subset of a set of target identifiers, wherein each target identifier in the set of target identifiers is related with only one computational resource, wherein each target identifier in the subset of target identifiers identifies the computational resource, and wherein a size of the subset of target identifiers is proportional to a computational capacity of the computational resource.

Rostoker et al. clearly does not disclose nor suggest most of the features of the dependent claims, notwithstanding the arguments in the Office action. The rejection repeatedly refers to the same section of Rostoker et al. for support, but as should be apparent with reference to this section that was copied hereinabove, Rostoker et al. does not disclose nor suggest the claimed features of the present invention.

With respect to the other claims that were rejected under Rostoker et al., independent claim 9 is directed to "a method in a data processing system for mapping a source identifier to a target identifier", and independent claim 42 is directed to a similar method; independent claims 20 and 58 and their dependent claims are directed to corresponding apparatuses, and independent claims 31 and 74 and their dependent claims are directed to corresponding computer program products. The Office action presents similar arguments in rejecting these claims, and

Applicant asserts that the argument that is provided hereinabove for independent claim 9 and its dependent claims are applicable to the other claims.

Examiner bears the burden of establishing a *prima facie* case of obviousness

The examiner bears the burden of establishing a *prima facie* case of obviousness based on the prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). Only when a *prima facie* case of obviousness is established does the burden shift to the applicant to produce evidence of nonobviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Rijckaert*, 9 F.3d 1531, 1532, 28 U.S.P.Q.2d 1955, 1956 (Fed. Cir. 1993). If the Patent Office does not produce a *prima facie* case of unpatentability, then without more the applicant is entitled to grant of a patent. *In re Oetiker*, 977 F.2d 1443, 1445, 24 U.S.P.Q.2d 1443, 1444 (Fed. Cir. 1992); *In re Grabiak*, 769 F.2d 729, 733, 226 U.S.P.Q. 870, 873 (Fed. Cir. 1985). In response to an assertion of obviousness by the Patent Office, the applicant may attack the Patent Office's *prima facie* determination as improperly made out, present objective evidence tending to support a conclusion of nonobviousness, or both. *In re Fritch*, 972 F.2d 1260, 1265, 23 U.S.P.Q.2d 1780, 1783 (Fed. Cir. 1992).

Rostoker et al. clearly fails to disclose or to suggest at least one feature of the present invention as recited within each independent claim, notwithstanding the arguments presented by the Office action, thereby rendering Rostoker et al. incapable of being used as primary reference as argued by the current rejection. Moreover, a hypothetical modification of Rostoker et al. would also fail to reach the claimed invention of the present patent application.

With respect to the claims of the present patent application, Applicant respectfully submits that it would not have been obvious for one having ordinary skill in the art to have used the applied prior art reference to reach the claimed invention. Hence, a rejection of the claims cannot be based upon the cited prior art to establish a *prima facie* case of obviousness. Therefore, a rejection of the claims under 35 U.S.C. § 103(a) has been shown to be insupportable in view of the cited prior art, and the claims are patentable over the applied reference. Applicant respectfully requests the withdrawal of the rejection of the claims.

IV. 35 U.S.C. § 103–Obviousness over O'Connell

Claims 1-8 are rejected under 35 U.S.C. § 103(a) as unpatentable over O'Connell et al., "Integrated data table in a network", U.S. Patent Number 6,661,787 B1, filed 04/06/1999, issued 12/09/2003. This rejection is traversed.

With respect to independent claims 1, 4, and 7, the rejection addresses each element of the claim only cursorily. For example, the rejection of claim 1, which is directed to a router, states that O'Connell et al. discloses "retrieving means", "reading means", and "modifying means". However, the rejection completely ignores significant features, specifically in the reading means, which is recited in claim 1 as follows:

> reading means for reading a target address from a table entry using the table index, wherein the target address has been related to and stored in the table entry based on a computed value from a relation computation using the table index and the target address as operands in the relation computation.

Similar features in the method steps of independent claims 4 and 7 are likewise ignored by the rejection. For example, claim 4 recites in its entirety:

4. A routing method in a data processing system comprising the steps of:

receiving a data packet;

retrieving a destination address from the data packet;

hashing the destination address to determine a table index into a table in a computer readable medium; and

reading a target address from a table entry using the table index, wherein the target address has been related to and stored in the table entry based on a computed value from a relation computation using the table index and the target address as operands in the relation computation;

modifying the data packet by storing the target address in the data packet as a next-hop destination address; and

transmitting the modified data packet.

Given the fact that the rejection of these claims has completely failed to address significant features of the claims, Applicant asserts that the Office action has failed to present a *prima facie* case of obviousness.

Not only has the rejection failed to present a proper obviousness argument, but more importantly, O'Connell et al. clearly fails to disclose or to suggest at least one feature of the present invention as recited within each independent claim, thereby rendering O'Connell et al. incapable of being used as primary reference as argued by the current rejection.

With respect to the claims of the present patent application, Applicant respectfully submits that it would not have been obvious for one having ordinary skill in the art to have used the applied prior art reference to reach the claimed invention. Hence, a rejection of the claims cannot be based upon the cited prior art to establish a *prima facie* case of obviousness. Therefore, a rejection of the claims under 35 U.S.C. § 103(a) has been shown to be insupportable in view of the cited prior art, and the claims are patentable over the applied reference. Applicant respectfully requests the withdrawal of the rejection of the claims.

V. Conclusion

It is respectfully urged that the present patent application is patentable, and Applicant kindly requests a Notice of Allowance.

For any other outstanding matters or issues, the examiner is urged to call or fax the below-listed telephone numbers to expedite the prosecution and examination of this application.

DATE: September 13, 2004

Respectfully submitted,

Joseph R. Burwell

Joseph R. Burwell
Reg. No. 44,468
ATTORNEY FOR APPLICANT

Law Office of Joseph R. Burwell
P.O. Box 28022
Austin, Texas 78755-8022
Voice: 866-728-3688 (866-PATENT8)
Fax: 866-728-3680 (866-PATENT0)
Email: joe@burwell.biz